

# Une approche PnP distribuée pour l'échantillonnage de problèmes inverses en grande dimension

Maxime Bouton\*, P.-A. Thouvenin\*, A. Repetti†‡ et P. Chainais\*

\*Université de Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

†School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK

‡Maxwell Institute for Mathematical Sciences, Edinburgh EH8 9BT, UK

## 1. Contexte et enjeux

**Problèmes inverses** : estimer  $\bar{\mathbf{x}} \in \mathbb{R}^N$  à partir d'observations  $\mathbf{y} \in \mathbb{R}^M$

$$\mathbf{y} = \mathcal{A}(\mathbf{H}\bar{\mathbf{x}}) \rightsquigarrow \pi(\mathbf{x} | \mathbf{y}) \propto \exp(-\phi_{\mathbf{y}}(\mathbf{H}\mathbf{x})) p(\mathbf{x})$$

$\mathbf{H}, \mathcal{A}$  modèles d'acquisition et de bruit  
avec  $\phi_{\mathbf{y}}$  attache aux données  
 $p$  *a priori* sur  $\mathbf{x}$

Mesure d'incertitude  $\Rightarrow$  Monte-Carlo par Chaînes de Markov (MCMC)

Qualité d'estimation  $\Rightarrow$  *a priori* appris, encodé par un réseau de neurones

Problèmes de très grandes dimensions ( $M \approx N > 10^6$ )

$\Rightarrow$  implémentation distribuée (architecture SPMD) [1]:

⊕ décomposer le problème sur  $B$  agents

⚠ limiter les communications

## 2. Approche proposée

Basée sur PnP-ULA [2] :

$$\begin{aligned} \mathbf{z}^{(t+1)} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \delta \mathbf{H}^* \nabla \phi_{\mathbf{y}}(\mathbf{H}\mathbf{x}^{(t)}) + \sqrt{2\delta} \mathbf{z}^{(t+1)} \\ &\quad + \frac{\alpha\delta}{\epsilon} (\mathbf{D}_{\epsilon}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}) + \frac{\delta}{\lambda} (\Pi_{\mathcal{C}}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}), \end{aligned}$$

avec  $\mathbf{D}_{\epsilon}$  débruiteur (profond) gaussien, variance  $\epsilon$   
 $\Pi_{\mathcal{C}}$  projecteur sur  $\mathcal{C}$

Propriété:  $f$  local  $\Leftrightarrow [f(\mathbf{x})]_i = \tilde{f}_i(\mathbf{x}_{[i]}), \quad \mathbf{x}_{[i]}$ : éléments voisins

Hypothèses :

- $\phi_{\mathbf{y}}$  additivement séparable  $\rightsquigarrow (\mathbf{y}_b)_{1 \leq b \leq B} \perp\!\!\!\perp \mathbf{x}$
- $\nabla \phi_{\mathbf{y}}, \Pi_{\mathcal{C}}$  et  $\mathbf{D}_{\epsilon}$  locaux  $\rightsquigarrow (\mathbf{x}_b)_{1 \leq b \leq B}$  + communications limitées

Choix de  $\mathbf{D}_{\epsilon}$  : Deep Dual Forward-Backward [3]

- Convolutions  $\Rightarrow$  Local ✓
- Peu de couches  $\Rightarrow$  Communications ↘

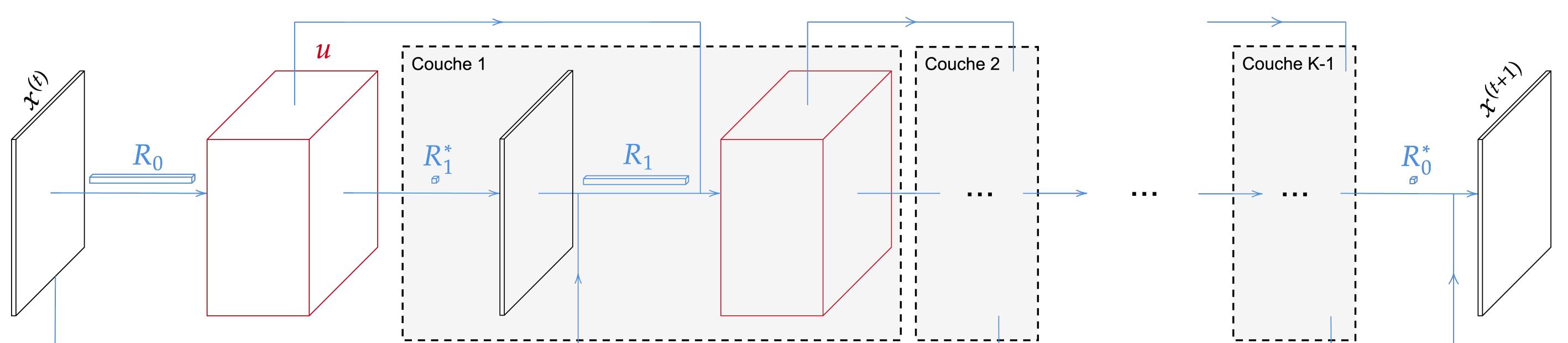
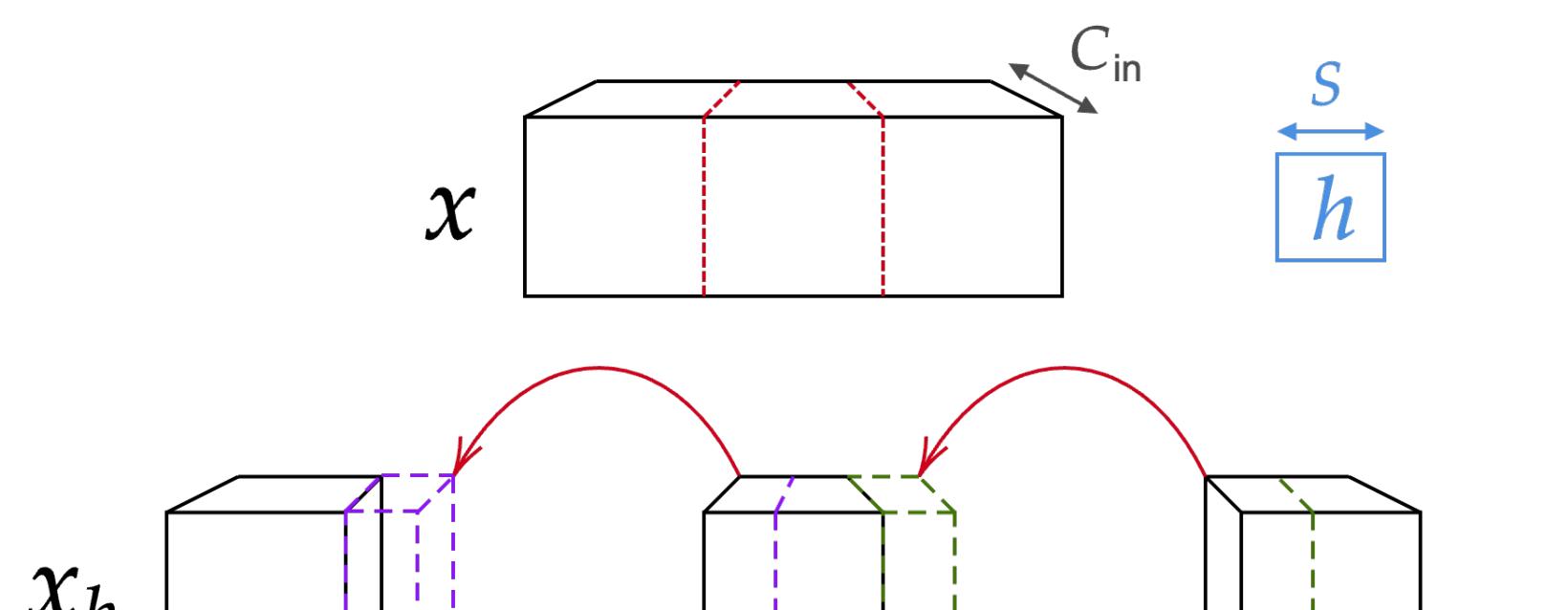


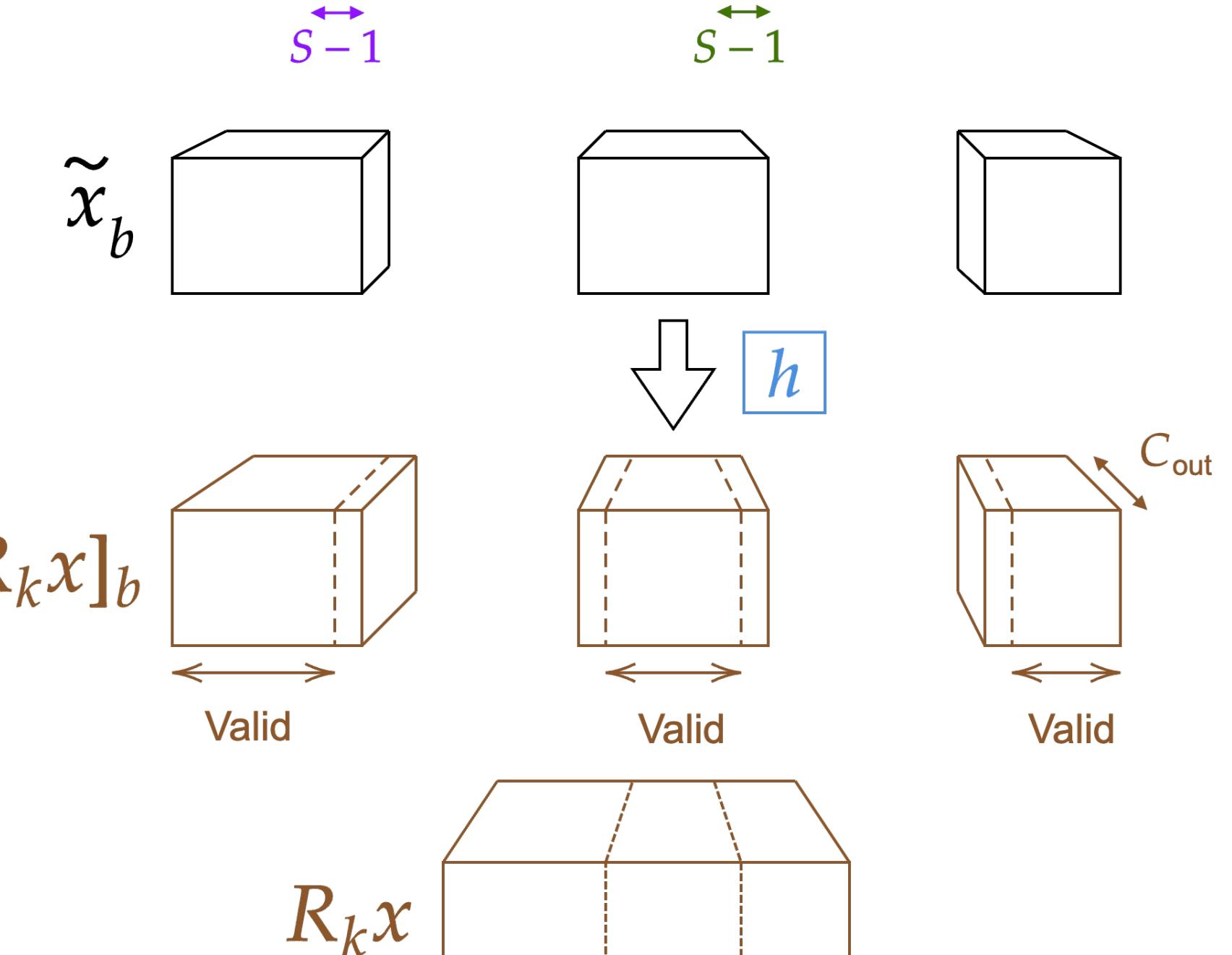
Figure 1: Structure du DDFB

Distribuer une convolution du DDFB :

1 Communications



2 Application de l'opérateur



3 Gestion effets de bord

## 3. Algorithme distribué proposé

Entrées:  $\mathbf{y} \in \mathbb{R}^M, \epsilon > 0, \alpha > 0, \lambda, \delta$

```

1 Pour chaque agent  $b \in \{1, \dots, B\}$  faire en parallèle
2 Charger et stocker  $\mathbf{y}_b \in \mathbb{R}^{M_b}$ 
3 Initialiser  $\mathbf{x}_b^{(0)} \in \mathbb{R}^{N_b}$ 
4 Pour  $t = 0$  à  $T - 1$  faire
    // Communiquer à chaque couche du DDFB pour calculer
     $d_b^{(t)} = [\mathbf{D}_{\epsilon}(\mathbf{x}^{(t)})]_b$ 
    // Mise à jour locale
     $\mathbf{g}_b^{(t)} = \frac{1}{\sigma^2} \mathbf{H}_b^* (\mathbf{H}_b \mathbf{x}_b^{(t)} - \mathbf{y}_b)$ 
    Tirer  $\mathbf{z}_b^{(t+1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{N_b})$ 
     $\mathbf{x}_b^{(t+1)} = \mathbf{x}_b^{(t)} - \delta \mathbf{g}_b^{(t)} + \frac{\alpha\delta}{\epsilon} (\mathbf{d}_b^{(t)} - \mathbf{x}_b^{(t)}) + \frac{\delta}{\lambda} (\Pi_{\mathcal{C}_b}(\mathbf{x}_b^{(t)}) - \mathbf{x}_b^{(t)}) + \sqrt{2\delta} \mathbf{z}_b^{(t+1)}$ 

```

Sorties:  $(\mathbf{x}^{(t)})_{1 \leq t \leq T}$

## 4. Expériences sur données synthétiques

Application: inpainting image couleur 80%, bruit gaussien  $\sigma = 0.03$   
Comparaison: PnP-FB utilisant DRUNet (séquentiel)



Figure 2: Ligne 1: Observations, vérité terrain, initialisation. Ligne 2: MAP (PnP-FB), MMSE (DPnP-ULA,  $B = 4$ ) et intervalle de crédibilité associé.

$B$	Temps / iter (ms)	Accélération	rSNR	SSIM
1	$199 \pm 0.32$	1	16.29	0.78
2	$137 \pm 0.45$	1.45	16.33	0.78
4	$105 \pm 0.94$	1.90	16.30	0.78

Table 1: Scalabilité forte

## 5. Conclusions

Échantillonneur PnP-ULA distribué

- conservation des garanties théoriques;
- implémentation multi-GPU;
- passage à l'échelle;
- contraintes sur le débruiteur;

- [1] P.-A. Thouvenin, A. Repetti, and P. Chainais, "A distributed split-Gibbs sampler with hypergraph structure for high-dimensional inverse problems," *J. Comput. and Graph. Stat.*, vol. 33, no. 3, pp. 814–832, 2024.
- [2] R. Laumont et al., "Bayesian imaging using Plug & Play priors: When Langevin meets Tweedie," *SIAM Journal on Imaging Sciences*, vol. 15, no. 2, pp. 701–737, 2022. arXiv: 2103.04715 [cs, eess, math, stat].
- [3] H. T. V. Le, A. Repetti, and N. Pustelnik, "Unfolded Proximal Neural Networks for Robust Image Gaussian Denoising," *IEEE Trans. Image Process.*, vol. 33, pp. 4475–4487, 2024.